What is Claimed is:

Patent Application No. 1:

1. A system for transmitting and receiving TCP/IP data packets using a hardware engine, comprising:

an inbound MAC Receive state machine for processing MAC frames received from a network;

an inbound IP verifier state machine for verifying IP packet headers

an inbound IP fragment processing state machine for processing and reassembling IP fragments; and

an inbound TCP state machine for processing TCP segments received from an IP layer.

2. The system of Claim 1 further comprising:

an outbound MAC Transmit state machine that sends MAC frames to a network;

an outbound IP state machine that processes IP data to be passed to a MAC layer for transmission; and

an outbound TCP state machine that processes TCP data to be passed to the IP layer for transmission.

3. The system of Claim 1, wherein the outbound IP state machine builds IP header data and passes the header data to the outbound MAC Transmit state machine.

4. The system of Claim 1, wherein the outbound TCP state machine builds TCP header data and passes the header data to the outbound IP state machine.

5. The system of Claim 1, wherein the inbound IP verifier state machine passes non-IP data packets to a host.

6. The system of Claim 1, wherein the inbound IP verifier state machine verifies IP packet header information and if the

header information is valid, then temporarily stores the packet in an external memory.

7. The system of Claim 1, wherein the inbound IP verifier state machine passes complete IP datagrams to the host that are non-TCP packets.

8. The system of Claim 1, wherein the inbound IP fragment processing state machine provides a timer to time each datagram reassembly with a programmable timer value.

9. The system of Claim 1, wherein the inbound TCP state machine maintains a segment re-assembly list for each network connection that is linked to a network control block and is used to re-order out of order TCP data segments.

10. The system of Claim 1, wherein the inbound TCP state machine passes in-order TCP segments to the host or to an upper layer protocol processor.

11. A system for processing network data packets using a hardware engine, comprising:

a verification module that verifies incoming data packets;

a first in-bound TCP processor for processing TCP segments received from a network;

a fragment processor that receives data packet fragments and reassembles the data packet fragments into complete datagrams for delivery; and

a second in-bound processor for processing incoming TCP segments destined for iSCSI.

12.The system of Claim 11, further comprising:

a first outbound processor that processes TCP data that is sent to a network; and

a second outbound processor that processes MAC and IP transfers to the network.

13. The system of Claim 12, wherein the second outbound processor also acts as a pass through processor for TCP data processed by the first outbound processor.

14. The system of Claim 12, wherein the first outbound processor builds TCP header data and passes the header data to the second outbound processor.

15. The system of Claim 11, wherein the verification module passes non-IP data packets to a host.

16. The system of Claim 11, wherein the verification module verfifies IP data packet header information and if the header information is valid, then the data packet is added to a list maintained by the verification module.

17. The system of Claim 11, wherein the fragment processor provides a timer to time each datagram reassembly with a programmable default timer value.

18. The system of Claim 11, wherein the fragment processor sets a flag if overlapping datagrams are received and the flag indicates when a TCP checksum must be re-run.

19. The system of Claim 11, wherein the first inbound processor re-orders out of order data segments.

20. The system of Claim 11, wherein the first inbound processor maintains a segment re-assembly list for each network connection and is linked with a network control block.

21. The system of Claim 11, wherein the first inbound processor includes a receive block that receives data; a

validation block that validates data segments; and an option block for validating a TCP timestamp that is found in TCP option data; and acknowledgement processor that performs TCP acknowledgement processing.

22. The system of Claim 11, wherein the first inbound processor receives a data segment with a TCP header and option data.

23. A system for processing incoming TCP data packets, comprising:

a input processing module that determines if a TCP connection is established and checks for TCP flags to determine if a TCP datapacket should be processed;

an acknowledgement processor module that handles any acknowledgement information included in the TCP packet;and

a Data processor module that handles any data included in the TCP data packet.

24. The system of Claim 23, wherein the first input processing module validates and saves TCP timestamps by checking if a received timestamp is greater than a most recently saved timestamp.

25. The system of Claim 23, wherein the Data Processor module determines if the received packet was in order or out of order and trims the packet if it requires trimming.

26. The system of Claim 23, wherein TCP connection state is organized in network control blocks and stored in a local memory.

27. A network control block (NCB) used in a system for processing network data packets using a hardware engine, comprising:

plural status flags, control flags, destination address, header fields and/or TCP connection information, wherein NCBs are used to provide plural parameters to plural modules in the system and are maintained in a local memory and/or host memory.

28. The NCB of Claim 27, wherein a copy of a NCB is maintained in a local memory for a TCP connection to process a TCP packet, without access to host memory.